# Algorithms for New Types of Fair Stable Matchings

**Frances Cooper**
School of Computing Science, University of Glasgow, UK
https://www.francescooper.net
f.cooper.1@research.gla.ac.uk

**David Manlove**
School of Computing Science, University of Glasgow, UK
http://www.dcs.gla.ac.uk/~davidm
david.manlove@glasgow.ac.uk

## Abstract

We study the problem of finding "fair" stable matchings in the *Stable Marriage problem with Incomplete lists* (SMI). For an instance $I$ of SMI there may be many stable matchings, providing significantly different outcomes for the sets of men and women. We introduce two new notions of fairness in SMI. Firstly, a *regret-equal stable matching* minimises the difference in ranks of a worst-off man and a worst-off woman, among all stable matchings. Secondly, a *min-regret sum stable matching* minimises the sum of ranks of a worst-off man and a worst-off woman, among all stable matchings. We present two new efficient algorithms to find stable matchings of these types. Firstly, the *Regret-Equal Degree Iteration Algorithm* finds a regret-equal stable matching in $O(d_0nm)$ time, where $d_0$ is the absolute difference in ranks between a worst-off man and a worst-off woman in the man-optimal stable matching, $n$ is the number of men or women, and $m$ is the total length of all preference lists. Secondly, the *Min-Regret Sum Algorithm* finds a min-regret sum stable matching in $O(d_sm)$ time, where $d_s$ is the difference in the ranks between a worst-off man in each of the woman-optimal and man-optimal stable matchings. Experiments to compare several types of fair optimal stable matchings were conducted and show that the Regret-Equal Degree Iteration Algorithm produces matchings that are competitive with respect to other fairness objectives. On the other hand, existing types of "fair" stable matchings did not provide as close an approximation to regret-equal stable matchings.

**2012 ACM Subject Classification** Theory of computation → Design and analysis of algorithms

**Keywords and phrases** Stable marriage, Algorithms, Optimality, Fair stable matchings, Regret-equality, Min-regret sum

**Digital Object Identifier** 10.4230/LIPIcs.SEA.2020.20

**Related Version** arxiv.org/abs/2001.10875

**Supplementary Material** zenodo.org/record/3630383 and zenodo.org/record/3630349

**Funding** *Frances Cooper*: Supported by an Engineering and Physical Sciences Research Council Doctoral Training Account EP/N509668/1
*David Manlove*: Supported by Engineering and Physical Sciences Research Council grant EP/P028306/1

## 1 Introduction

### 1.1 Background

The Stable Marriage problem (SM) was first introduced by Gale and Shapley [5] in their seminal paper "College Admissions and the Stability of Marriage", and comprises a set of men and a set of women, where each man has a strict preference over all women and vice versa. A *matching* in this setting is an assignment of men to women such that no man or woman is multiply assigned. A *stable matching* is then a matching in which there is no man-woman pair who would rather be assigned to each other than to their assigned partners.

In this paper we study an extension of SM, known as the Stable Marriage problem with Incomplete lists (SMI). An instance $I$ of SMI comprises two sets of agents, men $U = \{m_1, m_2, ..., m_n\}$ and women $W = \{w_1, w_2, ..., w_n\}$. Each man (woman) ranks a subset of women (men) in strict preference order. Let $m$ be the total length of all preference lists. A man $m_i$ finds a woman $w_j$ *acceptable* if $w_j$ appears on $m_i$'s preference list. Similarly, a woman $w_j$ finds a man $m_i$ *acceptable* if $m_i$ appears on $w_j$'s preference list. A pair $(m_i, w_j)$ is *acceptable* if $m_i$ finds $w_j$ acceptable and $w_j$ finds $m_i$ acceptable. A *matching* $M$ in this context is an assignment of men to women comprising acceptable pairs such that no man or woman is assigned to more than one person. Given a matching $M$, denote by $M(m_i)$ the woman $m_i$ is assigned to in $M$ (or if $m_i$ is unassigned then $M(m_i)$ is undefined); the notation $M(w_j)$ is defined similarly for a woman $w_j$. A pair $(m_i, w_j)$ is a *blocking pair* if 1) $(m_i, w_j)$ is an acceptable pair, 2) $m_i$ is unmatched or prefers $w_j$ to $M(m_i)$, and 3) $w_j$ is unmatched or prefers $m_i$ to $M(w_j)$. Matching $M$ is *stable* if it has no blocking pair.

In SMI, a stable matching always exists, and may be found in linear time using the Man-oriented Gale-Shapley Algorithm or the Woman-oriented Gale-Shapley Algorithm [5]. The Man-oriented Gale-Shapley Algorithm produces the *man-optimal* stable matching, that is, the unique stable matching in which each man is assigned their most-preferred woman in any stable matching. Unfortunately, the man-optimal stable matching is also *woman-pessimal* i.e., each woman is assigned their least-preferred man in any stable matching. Similarly the Woman-oriented Gale-Shapley Algorithm produces the woman-optimal (man-pessimal) stable matching.

Let $I$ be an instance of SMI and $n$ be the number of men or women in $I$. Let $\mathcal{M}$ be the set of all stable matchings in $I$, which may be exponential in size [10]. We note that by the *"Rural Hospitals" Theorem* [6], the same set of men and women are assigned in all stable matchings of $\mathcal{M}$. Thus in order to simplify future descriptions, we are able to use the Man-oriented Gale-Shapley Algorithm to find and remove all unassigned men and women from $I$ prior to any other operation. Without loss of generality, we assume that from this point onwards, all men and women in $I$ are assigned in any stable matching of $I$.

For an instance of SMI, it is natural to wish to find a stable matching in $\mathcal{M}$ which is in some sense fair for both sets of men and women. The *rank* of $m_i$ with respect to $M$ is defined as the location of $M(m_i)$ on $m_i$'s preference list, and is denoted $\text{rank}(m_i, M(m_i))$. An analogous definition of $\text{rank}(w_j, M(w_j))$ holds for a woman $w_j$. We define the *man-degree* $d_U(M)$ of $M$ as the largest rank of all men in $M$, that is, $d_U(M) = \max\{\text{rank}(m_i, M(m_i)) : m_i \in U\}$. Again an analogous definition of $d_W(M)$ holds for women. Define the *degree pair* of $M$, denoted $d(M) = (a, b)$ as the tuple of man- and woman-degrees in $M$, where $a = d_U(M)$ and $b = d_W(M)$. The *man-cost* $c_U(M)$ of matching $M$ is defined as the sum of ranks of all men, that is, $c_U(M) = \sum_{m_i \in U} \text{rank}(m_i, M(m_i))$. A similar definition of $c_W(M)$ holds for women. Finally, the *degree* of a matching $M$ is given by $d(M) = \max\{d_U(M), d_W(M)\}$ and the *cost* of matching $M$ is given by $c(M) = c_U(M) + c_W(M)$.

We now define four notions of fairness in the SMI context. Given a stable matching $M$, define its *balanced score* to be $\max\{c_U(M), c_W(M)\}$. $M$ is *balanced* [4] if it has minimum balanced score over all stable matchings in $\mathcal{M}$. Feder [4] showed that the problem of finding a balanced stable matching in SMI is NP-hard, although can be approximated within a factor of 2. This approximation factor was improved to $2 - \frac{1}{l}$, where $l$ is the length of the longest preference list, by Eric McDermid as noted in Manlove [14, pg. 110]. Gupta et al. [7]

showed that a balanced stable matching can be found in $O(f(n)8^t)$ time when parameterised by $t = k - \min\{c_U(M_0), c_W(M_z)\}$, where $f(n)$ is a function polynomial in $n$ and $k$ is the balanced score. The *sex-equal score* of $M$ is defined to be $|c_U(M) - c_W(M)|$. $M$ is *sex-equal* [9] if it has minimum sex-equal score over all stable matchings in $\mathcal{M}$. Finding a sex-equal stable matching was shown to be NP-hard by Kato [12]. This result was later strengthened by McDermid and Irving [15] who showed that, even in the case when preference lists have length at most 3, the problem of deciding whether there is a stable matching with sex-equal score 0 is NP-complete. Additionally, a polynomial-time algorithm to find a sex-equal stable matching is described for instances in which men have preference lists of length at most 2 (women's preference lists remaining unbounded) [15]. A stable matching $M$ is *egalitarian* [13] if $c(M)$ is minimum over all stable matchings in $\mathcal{M}$, and may be found in $O(m^{1.5})$ time [4]. Finally, a stable matching $M$ is *minimum regret* [13] if $d(M)$ is minimum among all stable matchings in $\mathcal{M}$. It is possible to find a minimum regret stable matching in $O(m)$ time [8]. These definitions of fairness are summarised in Table 1.

**Table 1** Commonly used definitions of fair stable matchings in SMI. Our contributions are labelled with an *.

| | Cost | Degree |
|---|---|---|
| Minimising the maximum | $\min_{M \in \mathcal{M}} \max\{c_U(M), c_W(M)\}$<br>Balanced stable matching [4] | $\min_{M \in \mathcal{M}} \max\{d_U(M), d_W(M)\}$<br>Minimum regret stable matching [13] |
| Minimising the absolute difference | $\min_{M \in \mathcal{M}} \lvert c_U(M) - c_W(M)\rvert$<br>Sex-equal stable matching [9] | $\min_{M \in \mathcal{M}} \lvert d_U(M) - d_W(M)\rvert$<br>Regret-equal stable matching * |
| Minimising the sum | $\min_{M \in \mathcal{M}} (c_U(M) + c_W(M))$<br>Egalitarian stable matching [13] | $\min_{M \in \mathcal{M}} (d_U(M) + d_W(M))$<br>Min-regret sum stable matching * |

In Table 1 there are two new natural definitions of fairness that can be studied.

- We define the *regret-equality score* $r(M)$ as $|d_U(M) - d_W(M)|$ for a given stable matching $M$. $M$ is *regret-equal* if $r(M)$ is minimum, taken over all stable matchings in $\mathcal{M}$. Note that in general we will prefer a regret-equal stable matching $M$ such that $d_U(M) + d_W(M)$ is minimised (e.g. $d(M) = (3, 3)$ rather than $d(M) = (10, 10)$).
- We define the *regret sum* as $d_U(M) + d_W(M)$ for a given stable matching $M$. $M$ is *min-regret sum* if $d_U(M) + d_W(M)$ is minimum taken over all stable matchings in $\mathcal{M}$.

## 1.2 Motivation

Matching algorithms are widely used in the real world to solve allocation problems based on SMI and its variants. A famous example of this is the *National Resident Matching Program* (NRMP). This scheme has been running in the US since 1952, and involves the allocation of thousands of graduating medical students to hospitals [16]. Other matching schemes involve the allocation of students to projects [1] and the allocation of kidney donors to kidney patients [2].

Let mentees take the place of men and mentors take the place of women. Thus, mentees (mentors) rank a subset of mentors (mentees) and may only be allocated one mentor (mentee) in any matching. If we used the (renamed) Mentee-Oriented Gale-Shapley Algorithm [5] to find a stable matching of mentees to mentors, then we would find a mentee-optimal stable matching $M$. However, as previously discussed, this would also be a mentor-pessimal stable matching. A similar but reversed situation happens using the (also renamed) Mentor-Oriented Gale-Shapley Algorithm [5]. Therefore we may wish to find a stable matching that is in some sense fair between mentees and mentors using some of the criteria described in the previous section. All the types of fair stable matchings described in Table 1 are viable candidates. However, as previously described, each of the problems of finding a balanced stable matching or a sex-equal stable matching is NP-hard, and there are existing polynomial time algorithms in the literature to find only two types of fair stable matchings, namely an egalitarian stable matching (in $O(m^{1.5})$ time) [4] and a minimum regret stable matching (in $O(m)$ time) [8]. Therefore, additional definitions of new, fair stable matchings and polynomial-time algorithms to calculate them provide additional choice for a matching scheme administrator.

Moreover, we may be interested in finding a measure that gives a worst-off mentee a partner of rank as close as possible to that of a worst-off mentor. However, from our experimental work in Section 5, we found that there was no other type of optimal stable matching that closely approximates the regret-equality score of the regret-equal stable matching. Indeed, results show that there exist regret-equal stable matchings with balanced score, cost and degree that are close to that of a balanced stable matching, an egalitarian stable matching and a minimum regret stable matching, respectively. This motivates the search for efficient algorithms to produce a regret-equal stable matching that has "good" measure relative to other types of fair stable matching.

Whilst the practical motivation for studying min-regret sum stable matchings may not be as strong as in the regret-equality case, theoretical motivation comes from completing the study of the algorithmic complexity of computing all types of fair stable matchings relative to cost and degree, as shown in Table 1.

## 1.3 Contribution

In this paper, we present two efficient algorithms: one to find a regret-equal stable matching, and one to find a min-regret sum stable matching, in an instance $I$ of SMI. Let $M_0$ and $M_z$ be the man-optimal and woman-optimal stable matchings in $I$. First we present the *Regret-Equal Degree Iteration Algorithm* (REDI), to find a regret-equal stable matching in an instance $I$ of SMI, with time complexity $O(d_0nm)$, where $d_0 = |d_U(M_0) - d_W(M_0)|$. This is the main result of the paper. Second we present the *Min-Regret Sum Algorithm* (MRS), to find a min-regret sum stable matching in an instance $I$ of SMI, with time complexity $O(d_sm)$, where $d_s = d_U(M_z) - d_U(M_0)$. In addition to this theoretical work, the REDI algorithm was implemented and its performance was compared against an algorithm to enumerate all stable matchings [8] (exponential in the worst case). Finally, experiments were conducted to compare six different types of optimal stable matchings (balanced, sex-equal, egalitarian, min-regret, regret-equal, min-regret sum), and output from Algorithm REDI, over a range of measures (including balanced score, sex-equal score, cost, degree, regret-equality score, regret sum). In addition to the observations already discussed in Section 1.2, we found a large variation in sex-equal scores and regret-equality scores among the six different types of optimal stable matching, and, a far smaller variation for the balanced score, cost, degree and regret sum measures. This smaller variation also includes outputs of Algorithm REDI, indicating that we are able to find a regret-equal stable matching in polynomial time with a likely good balanced score, cost and degree using this algorithm. Indeed, we find in practice that Algorithm REDI approximates these types of optimal stable matchings at an average of 9.0%, 1.1% and 3.0% over their respective optimal values, for randomly-generated instances with $n = 1000$.

### 1.4 Structure of the paper

Section 2 describes a *rotation* and related concepts in SMI that will be used later in the paper. Sections 3 and 4 describe Algorithm REDI and Algorithm MRS respectively, giving in each case pseudocode, correctness proofs and time complexity calculations. An experimental evaluation is given in Section 5. Finally, future work is presented in Section 6.

## 2 Structure of stable matchings

For some stable matching $M$ in an instance $I$ of SMI, let $s(m_i, M)$ denote the next woman on $m_i$'s preference list (starting from $M(m_i)$) who prefers $m_i$ to $M(s(m_i, M))$ (their partner in $M$). A *rotation* $\rho$ is then a sequence of man-woman pairs $\{(m_1, w_1), (m_2, w_2), ..., (m_q, w_q)\}$ in $M$, such that $m_{i+1} = M(s(m_i, M))$ for $1 \leq i \leq q$ where addition is taken modulo $q$ [11]. We say rotation $\rho$ is *exposed* on $M$ if $\{(m_1, w_1), (m_2, w_2), ..., (m_q, w_q)\} \subseteq M$. If $\rho$ is exposed on $M$, we may *eliminate* $\rho$ on $M$, that is, remove all pairs of $\rho$ from $M$ and add pairs $(m_i, w_{i+1})$ for $1 \leq i \leq q$, where addition is taken modulo $q$, in order to produce another stable matching $M'$ of $I$. The *rotation poset* $R_p(I)$ of $I$ indicates the order in which rotations may be eliminated. Rotation $\rho$ is said to *precede* rotation $\tau$ if $\tau$ is not exposed until $\rho$ has been eliminated. There is a one-to-one correspondence between the set of stable matchings and the set of closed subsets of $R_p(I)$ [11, Theorem 3.1]. Gusfield and Irving [9] describe a graphical structure known as the *rotation digraph* $R_d(I)$ of $I$ which is based on $R_p(I)$ and allows for the enumeration of all stable matchings in $O(m + n|\mathcal{M}|)$ time.

Let $R$ be the set of rotations of $I$. Then $R_j(M)$ is the set of rotations that contain a women of rank $j$ in $M$, that is, $R_j(M) = \{\sigma \in R : (m, w) \in \sigma \wedge \text{rank}(w, M(w)) = j)\}$. Let $M_z$ be the woman-optimal stable matching [5]. For any stable pair $(m_i, w_j) \notin M_z$, let $\phi(m_i, w_j)$ denote the unique rotation containing pair $(m_i, w_j)$. Finally, denote by $c(\rho)$ the closure of rotation $\rho$ and similarly denote by $c(R')$ the closure of set of rotations $R'$. We say that the closure of an undefined rotation or an empty set of rotations is the empty set.

## 3 Algorithm to find a regret-equal stable matching in SMI

### 3.1 Description of the Algorithm

Algorithm REDI, which finds a regret-equal stable matching in a given instance $I$ of SMI, is presented as Algorithm 1. For an instance $I$ of SMI, Algorithm REDI begins with operations to find the man-optimal and woman-optimal stable matchings, $M_0$ and $M_z$, found using the Man-oriented and Women-oriented Gale-Shapley Algorithm [5]. The set of rotations $R$ is also found using the Minimal Differences Algorithm [11].

Let $d(M_0) = (a_0, b_0)$. If $a_0 = b_0$ then we must have an optimal stable matching and so we output $M_0$ on Line 5. If $a_0 > b_0$ then any other matching $M'$, where $d(M') = (a', b')$, must have $a' \geq a_0$ and $b' \leq b_0$ since any rotation (or combination of rotations) eliminated on the man-optimal matching $M_0$ will make men no better off and women no worse off. Therefore $M_0$ is optimal and so it is returned on Line 5. Now suppose $a_0 < b_0$. Throughout the algorithm we save the best matching found so far to the variable $M_{opt}$ starting with $M_0$.

We know that a matching exists with $d_0 = b_0 - a_0$ and so we try to improve on this, by finding a matching $M$ with $r(M) < d_0$.

We create several "columns" of possible degree pairs of a regret-equal matching as follows. The top-most pairs for columns $k \geq 1$ are given by the sequence

$$\big((a_0, b_0), (a_0 + 1, b_0), (a_0 + 2, b_0), ..., (\min\{n, 2b_0 - a_0 - 1\}, b_0)\big).$$

The sequence of pairs for column $k$ ($1 \leq k \leq \min\{2d_0, n - a_0 + 1\}$) from top to bottom is given by

$$\big((a_0+k-1, b_0), (a_0+k-1, b_0-1), (a_0+k-1, b_0-2), ..., (a_0+k-1, \max\{a_0-d_0+k, 1\})\big).$$

At this point as long as the size $n$ of the instance satisfies $n \geq 2b_0-a_0-1$ and $a_0-d_0+1 \geq 1$, the possible degree pairs of a regret-equal matching are shown in Figure 3 of [3]. We know this accounts for all possible degree pairs since, as above, if $M'$ is any matching not equal to $M_0$, where $d(M') = (a', b')$, it must be that $a' \geq a_0$ and $b' \leq b_0$. Setting $b' = b_0$, the largest $a'$ could be is given by $b_0$ added to the maximum possible improved difference $d_0 - 1$, that is, $a' = b_0 + d_0 - 1 = 2b_0 - a_0 - 1$. If $n < 2b_0 - a_0 - 1$ then we only consider the first $n - a_0 + 1$ columns in Figure 3 of [3]. The $a_0 - d_0 + k$ value is obtained by noting that if $x$ is the final value of women's degree for the column sequence above then $a_0 + k - 1 - x = d_0 - 1$ and so $x = a_0 + k - d_0$. Figure 4 of [3] shows an example of the possible regret-equal degree pairs when $d(M_0) = (2, 6)$ and $n \geq 9$.

The column operation (Algorithm 2) works as follows. Let local variable $M$ hold the current matching for this column, and let local variable $Q$ be the set of rotations corresponding to $M$. Iteratively we first test if $r(M) < r(M_{opt})$ setting $M_{opt}$ to $M$ if so. We now check whether $d_U(M) \geq d_W(M)$. If it is, then any further rotation for this column will only make $r(M)$ larger, and so we stop iterating for this column, returning $M_{opt}$. Next, we find the set of rotations $Q'$ in the closure of $R_b(M) \subseteq R$ that are not already eliminated to reach $M$. If eliminating these rotations would either increase the men's degree or not decrease the women's degree, then we return $M_{opt}$. Otherwise, set $M$ to be the matching found when eliminating these rotations.

If after the column operation, $d_U(M_{opt}) = d_W(M_{opt})$, then we have a regret-equal matching and it is immediately returned on Lines 10 or 24 of Algorithm 1.

The column operation described above is called first from the man-optimal stable matching $M_0$ on Line 8, to iterate down the first column. Then for each man $m_i$ we do the following. Let $M$ be set to $M_0$. Iteratively we eliminate $(m_i, M(m_i))$ from $M$ by eliminating rotation $\rho$ and its predecessors (not already eliminated to reach $M$) such that $(m_i, M(m_i)) \in \rho$. We continue doing this until both the men's degree increases and $\text{rank}(m_i, M(m_i)) = d_U(M)$ (in the same operation). This has the effect of jumping our focus from some column of possible degree pairs, to another column further to the right with $m_i$ being one of the lowest ranked men in $M$. Once we have moved to a new column we perform the column operation described above. If either $m_i$ has the same partner in $M$ as in $M_z$ (hence there are no rotations left that move $m_i$) or $d_U(M) > d_W(M)$ (further rotations will only increase the regret-equality score), then we stop iterating for $m_i$. In this case we restart this process for the next man, or return $M_{opt}$ if we have completed this process for all men. Note that since at the end of a while loop iteration, if $r(M) = 0$ then $M_{opt}$ is returned, it is not possible for the condition $d_U(M) = d_W(M)$ to ever be satisfied in the while loop clause.

**Algorithm 1** REDI($I$), returns a regret-equal stable matching for an instance $I$ of SMI.

**Require:** An instance $I$ of SMI.
**Ensure:** Return a regret-equal stable matching $M_{opt}$.

$M_0 \leftarrow$ MGS($I$) ▷ $M_0$ is the man-optimal stable matching found using the Man-oriented Gale-Shapley Algorithm (MGS) [5].
$M_z \leftarrow$ WGS($I$) ▷ $M_z$ is the woman-optimal stable matching found using the Woman-oriented Gale-Shapley Algorithm (WGS) [5].
$R \leftarrow$ MIN-DIFF($I$) ▷ $R$ is the set of rotations found using the Minimal Differences Algorithm (MIN-DIFF) [11].
**if** $d_U(M_0) \geq d_W(M_0)$ **then**
  **return** $M_0$
**end if**
$M_{opt} \leftarrow M_0$ ▷ $M_{opt}$ is the best stable matching found so far.
$M_{opt} \leftarrow$REDI-COL$(I, M_0, \emptyset, M_{opt})$ ▷ Find the best matching for the first column.
**if** $r(M_{opt}) = 0$ **then**
  **return** $M_{opt}$
**end if**
**for each** $m_i \in U$ **do** ▷ For each man.
  $M \leftarrow M_0$ ▷ $M$ is the matching we start from for $m_i$ at the beginning of each column.
  $Q \leftarrow \emptyset$ ▷ $Q$ is the set of rotations corresponding to $M$.
  **while** $(m_i, M(m_i)) \notin M_z$ **and** $d_U(M) < d_W(M)$ **do**
    $\rho = \phi(m_i, M(m_i))$
    $a \leftarrow d_U(M)$
    $Q' \leftarrow c(\rho) \backslash Q$
    $M \leftarrow M/Q'$ ▷ Rotations in $Q'$ are eliminated in order defined by the rotation poset of $I$.
    $Q \leftarrow Q \cup Q'$
    **if** $d_U(M) > a$ **and** rank$(m_i, M(m_i)) = d_U(M)$ **then** ▷ The men's degree has increased and $m_i$ is a worst ranked man in $M$.
      $M_{opt} \leftarrow$REDI-COL$(I, M, Q, M_{opt})$ ▷ Find the best matching for this column.
      **if** $r(M_{opt}) = 0$ **then**
        **return** $M_{opt}$
      **end if**
    **end if**
  **end while**
**end for**
**return** $M_{opt}$

## 3.2 Correctness proof and time complexity

In this section we state the correctness and time complexity results for Algorithm REDI. The proofs of these theorems may be found in [3, Appendix A.2].

▶ **Theorem 1.** *Let $I$ be an instance of SMI. Any matching produced by Algorithm REDI is a regret-equal stable matching of $I$.*

▶ **Theorem 2.** *Let $I$ be an instance of SMI. Algorithm REDI always terminates within $O(d_0 nm)$ time, where $d_0 = |d_U(M_0) - d_W(M_0)|$, $n$ is the number of men or women in $I$, $m$ is the total length of all preference lists and $M_0$ is the man-optimal stable matching.*

**Algorithm 2** REDI-COL($I, M, Q, M_{opt}$), subroutine for Algorithm 1. Column operation for the current column $d_U(M)$. Returns $M_{opt}$, the best stable matching found so far (according to the regret-equality score).

**Require:** An instance $I$ of SMI, stable matching $M$, the closure of $M$, $Q$ and $M_{opt}$ the best stable matching found so far (according to the regret-equality score).

**Ensure:** Finds the best stable matching (according to the regret-equality score) found when incrementally eliminating women of worst rank from the current matching, without increasing the men's degree. If an improvement is made then $M_{opt}$ is updated. $M_{opt}$ is returned. All variables used within Algorithm 2 are understood to be local.

```
a ← d_U(M)
while true do
    if r(M) < r(M_opt) then
        M_opt ← M
    end if
    if d_U(M) ≥ d_W(M) then    ▷ Further rotations for this column would only increase the difference in degree of men and women.
        return M_opt
    end if
    b ← d_W(M)
    Q' ← c(R_b(M))\Q
    if d_U(M/Q') > a ∨ d_W(M/Q') = b then
        return M_opt
    else
        M ← M/Q'    ▷ Rotations in Q' are eliminated in order defined by the rotation poset of I.
        Q ← Q ∪ Q'
    end if
end while
```

## 3.3 Regret-equal stable matchings with minimum cost

We may seek a regret-equal stable matching with minimum cost over all regret-equal stable matchings. This may be achieved in $O(nm^{2.5})$ time using the following process.

We define the *deletion* of pair $(m_i, w_j)$ as the removal of $w_j$ from $m_i$'s preference list and the removal of $m_i$ from $w_j$'s preference list. *Truncating men's preference lists at* $t$, where $1 \leq t \leq n$, is then the process of deleting pair $(m_i, w_j)$ for each $(m_i, w_j)$ such that $\text{rank}(m_i, w_j) > t$. An analogous definition holds for women. For a given SMI instance $I$, first find the regret-equality score $r$ of the regret-equal stable matching using Algorithm REDI in $O(d_0 nm)$ time. Then, iterate over all possible man-woman degree pairs $(a, b)$ such that $|a - b| = r$ (there are $O(n)$ such pairs). For each such degree pair $(a, b)$, truncate men at $a$ and women at $b$, creating instance $I'$. Then, for each of the $O(m)$ man-woman pairs $(m_i, w_j)$ in $I'$, fix $m_i$ with his $a$th-choice partner and $w_j$ with her $b$th-choice partner (where ranks are taken with respect to instance $I$), if possible. If this is not possible then continue to the next degree pair. Assume that $w'_j$ is $m_i$'s $a$th-choice partner, and $m'_i$ is $w_j$'s $b$th-choice partner. In $I'$, we now delete pairs $(m''_i, w''_j)$ for any $w''_j$ such that $m_i$ prefers $w''_j$ to $w'_j$ and $w''_j$ prefers $m_i$ to $m''_i$. Also delete the pair $(m''_i, w''_j)$ for any $m''_i$ such that $w_j$ prefers $m''_i$ to $m'_i$ and $m''_i$ prefers $w_j$ to $w''_j$. Next we delete all remaining preference list elements of $m_i$ except $w'_j$ and all remaining preference list elements of $w_j$ except $m'_i$. The Gale-Shapley Algorithm is run to check that a stable matching of size $n$ exists in $I'$. If no such stable matching exists then we move on to the next degree pair. Feder's Algorithm may then be used to find an egalitarian stable matching in the reduced SMI instance $I'$ in $O(m^{1.5})$ time (using the original ranks in $I$ as costs). This makes a total of $O(nm^{2.5})$ time to find a regret-equal stable matching with minimum cost.

## 4 Algorithm to find a min-regret sum stable matching in SMI

Algorithm MRS, which finds a min-regret sum stable matching, given an instance of SMI, is presented as Algorithm 3. First, the man-optimal and woman-optimal stable matchings, $M_0$ and $M_z$, are found using the Man-oriented and Women-oriented Gale-Shapley Algorithms [5]. The best matching found so far, denoted $M_{opt}$ is initialised to $M_0$. We then iterate over each possible man degree $a$ between $d_U(M_0)$ and $d_U(M_z)$ inclusive, where an improvement of $M_{opt}$, according to the regret sum, is still possible. As an example, suppose $M_{opt}$ has a regret sum of 5 with $d_U(M_{opt}) = 2$ and $d_W(M_{opt}) = 3$. Then, it is not worth iterating over any man degree greater than 3 since it will not be possible to improve on the regret sum of 5 by doing so. For each iteration of the while loop, we truncate the men's preference lists at $a$, and find the woman-optimal stable matching $M_z^T$ for this truncated instance. If the regret sum of $M_z^T$ is smaller than that of $M_{opt}$, we update $M_{opt}$ to $M_z^T$. After all iterations over possible men's degrees are completed, $M_{opt}$ is returned.

**Algorithm 3** MRS($I$), returns a min-regret sum stable matching for an instance $I$ of SMI.

**Require:** An instance $I$ of SMI.
**Ensure:** Return a min-regret sum stable matching $M_{opt}$.

$M_0 \leftarrow$ MGS($I$) $\triangleright$ $M_0$ is the man-optimal stable matching found using the Man-oriented Gale-Shapley Algorithm (MGS) [5].
$M_z \leftarrow$ WGS($I$) $\triangleright$ $M_z$ is the woman-optimal stable matching found using the Woman-oriented Gale-Shapley Algorithm (WGS) [5].
$M_{opt} \leftarrow M_0$
$a \leftarrow d_U(M_0)$
**while** $a \leq d_U(M_z)$ **and** $a + 1 < d_U(M_{opt}) + d_W(M_{opt})$ **do**
    $I_T \leftarrow$instance $I$ where men's preference lists are truncated at rank $a$.
    $M_z^T \leftarrow$ WGS($I_T$)
    **if** $d_U(M_z^T) + d_W(M_z^T) < d_U(M_{opt}) + d_W(M_{opt})$ **then**
        $M_{opt} \leftarrow M_z^T$
    **end if**
    $a \leftarrow a + 1$
**end while**
**return** $M_{opt}$

Let $d_s$ denote the difference between the degree of men in the woman-optimal stable matching $M_z$, and in the man-optimal stable matching $M_0$, that is $d_s = d_U(M_z) - d_U(M_0)$. Theorem 3 as follows states that Algorithm MRS produces a min-regret sum stable matching in $O(d_s m)$ time. See [3, Appendix B] for the proof of this Theorem.

▶ **Theorem 3.** *Let $I$ be an instance of SMI. Algorithm MRS produces a min-regret sum stable matching in $O(d_s m)$ time, where $d_s = d_U(M_z) - d_U(M_0)$, $m$ is the total length of all preference lists, and $M_0$ and $M_z$ are the man-optimal and woman-optimal stable matchings respectively.*

# 5 Experiments

## 5.1 Methodology

An Enumeration Algorithm (ENUM) exists to find the set of all stable matchings of an instance $I$ of SMI in $O(m + n|\mathcal{M}|)$ time [8]. Within this time complexity, it is possible to output a regret-equal stable matching from this set of stable matchings, by keeping track of the best stable matching found so far (according to the regret-equality score) as they are created. We randomly generated instances of SM, in order to compare the performance of Algorithms REDI and ENUM. Using output from Algorithm ENUM, we also investigated the effect of varying instance sizes, for six different types of optimal stable matchings (balanced, sex-equal, egalitarian, min-regret, regret-equal, min-regret sum), and also output from Algorithm REDI, over a range of measures (including balanced score, sex-equal score, cost, degree, regret-equality score, regret sum). Tests were run over 19 different instance types with varying instance size ($n \in \{10, 20, ..., 100, 200, ..., 1000\}$). All instances tested were complete with uniform distributions on preference lists. Experiments were run over 500 instances of each instance type.

Each instance was run over the two algorithms described above with a timeout time of 1 hour for each algorithm. No instances timed out for these experiments. Experiments were conducted on a machine running Ubuntu version 18.04 with 32 cores, 8×64GB RAM and Dual Intel® Xeon® CPU E5-2697A v4 processors. Instance generation, correctness and statistics summarisation programs, and plot and LaTeX table generation were all written in Python and run on Python version 3.6.1. All other code was written in Java and compiled using Java version 1.8.0. Each instance was run on a single thread with 16 instances run in parallel using GNU Parallel [17]. Serial Java garbage collection was used with a maximum heap size of 2GB distributed to each thread. Code and data repositories for these experiments can be found at zenodo.org/record/3630383 and zenodo.org/record/3630349 respectively. Comprehensive correctness testing was conducted, a description of which may be seen in [3, Appendix C.1].

## 5.2 Experimental results summary

Figure 1 shows a comparison of the time taken to execute the two algorithms over increasing values of $n$. Precise data for this plot can be seen in Table 2 of [3, Appendix C.2] which gives the mean, median, 5th percentile and 95th percentile durations for Algorithms REDI and ENUM. In Figure 1, the median values of time taken for each algorithm are plotted and a 90% confidence interval is displayed using the 5th and 95th percentile measurements. Additional experiments and evaluations not discussed here may also be found in [3, Appendix C.3].

Figure 2 shows comparisons of six different types of optimal stable matchings (balanced, sex-equal, egalitarian, min-regret, regret-equal, min-regret sum), and output from Algorithm REDI, over a range of measures (including balanced score, sex-equal score, cost, degree, regret-equality score, regret sum), as $n$ increases. Optimal stable matching statistics involving a measure determined by cost (respectively degree) are given a green (respectively blue) colour. For a particular fairness objective A and a particular fairness measure B, there may be a set of several stable matchings that are optimal with respect to A. In this case we choose a matching from this set that has best possible measure with respect to B. For example, if we are looking at the regret-equality score, for a particular instance, we find a sex-equal stable matching that has smallest regret-equality score (over the set of all sex-equal stable matchings) and use this value to plot the regret-equality score for this type of optimal stable matching. This process is replicated for the other types of optimal stable matching. In each case the mean measure value is plotted for the given type of optimal stable matching. Data for these plots may be found in Tables 3, 4, 5, 6, 7 and 8 of [3, Appendix C.2].

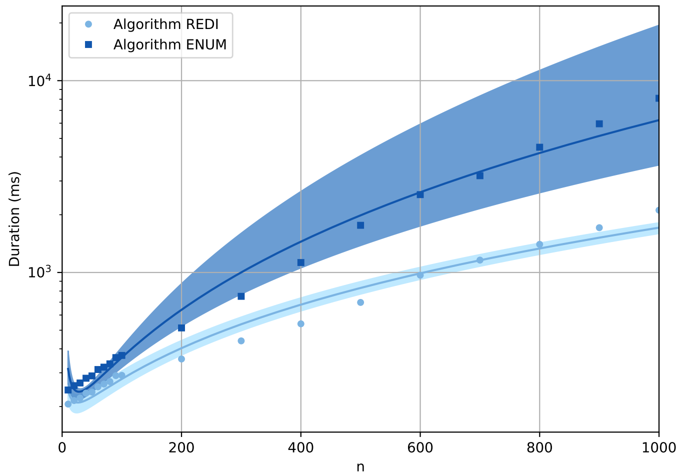
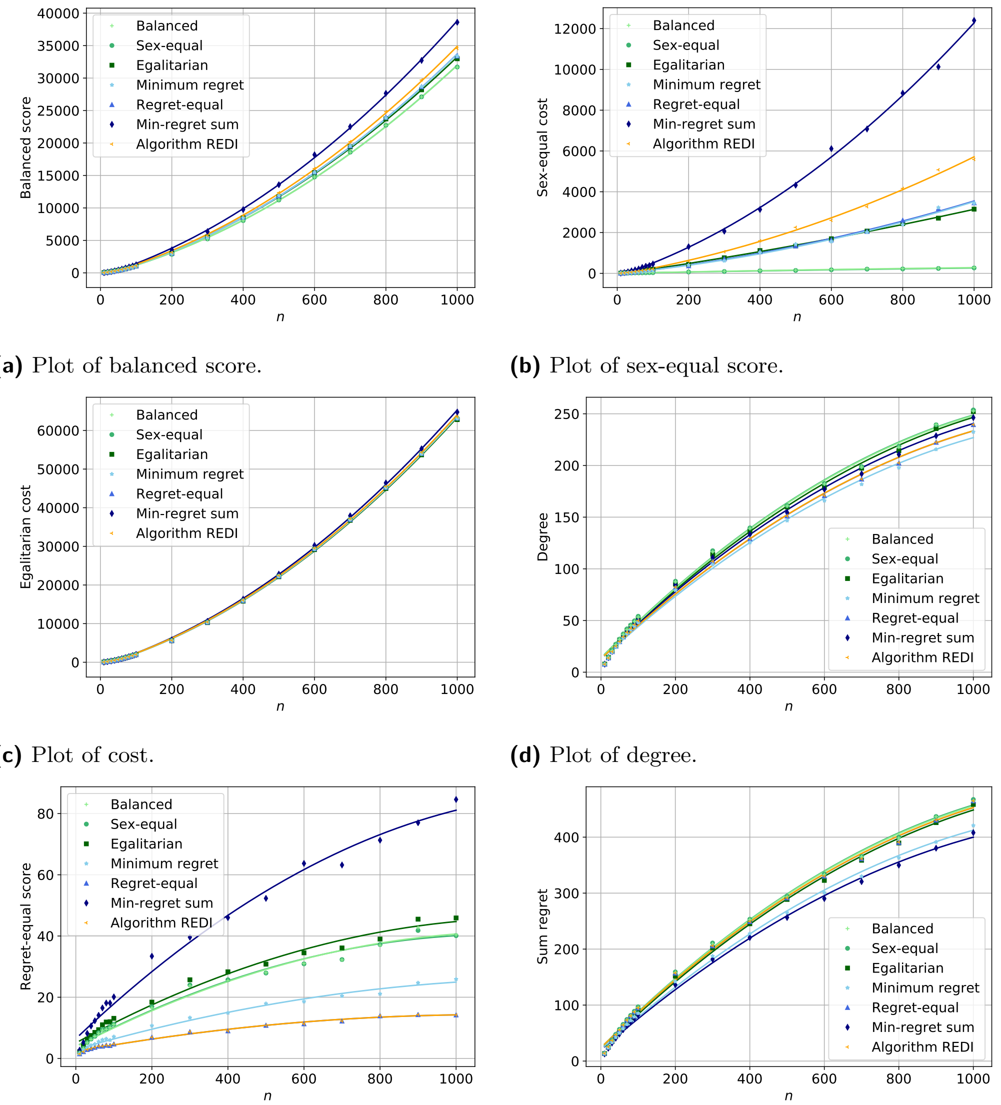



**Figure 1** A log plot of the time taken to execute Algorithms REDI and ENUM. A second order polynomial model has been assumed for best-fit lines.

The main results of these experiments are:

- *Time taken:* It is clear from Figure 1 in that Algorithm REDI is the faster algorithm in practice, taking approximately 2s to solve an instance of size $n = 1000$ with very little variation. In contrast, Algorithm ENUM takes around 8s for an instance of size $n = 1000$ with a far larger variation.
- *Sex-equal score:* A wide variation in sex-equal score over the six optimal matchings can be seen in Figure 2b (and Table 4 in [3]). Sex-equal and balanced stable matchings are extremely closely aligned giving a mean sex-equal score of 265.0 and 284.0 respectively for the instance type with $n = 1000$. Min-sum regret stable matchings, on the other hand, performed the least well with a mean sex-equal score of 12400.0 for the same instance type.
- *Regret-equality score:* Similar to the previous point we see a wide variation in regret-equality score over the six optimal stable matchings in Figure 2e (and Table 7 in [3]). For the instance type with $n = 1000$, this ranges from a mean regret-equality score of 14.2 for the regret-equal stable matching to 84.6 for the minimum regret stable matching. It is interesting to note that the type of optimal stable matching (out of the six optimal stable matchings tested) whose regret-equality score tends to be furthest away from that of a regret-equal stable matching is the min-regret sum stable matching. This may be due to the fact that minimising the sum of two measures does not necessarily force the two measures to be close together.
- *Output from Algorithm REDI:* Due to the wide variation of regret-equality scores among different types of optimal stable matchings (as described above) it is clear that no other optimal stable matching is able to closely approximate a regret-equal stable matching, which highlights the importance of Algorithm REDI that is designed specifically for optimising this measure. Interestingly, Algorithm REDI is also competitive in terms of balanced score, cost and degree. Indeed, we can see from Tables 3, 5 and 6 in [3], that

Algorithm REDI approximates these types of optimal stable matchings at an average of 9.0%, 1.1% and 3.0% over their respective optimal values, for instances with $n = 1000$. Over all instance sizes, these values are within ranges $[4.0\%, 10.9\%]$, $[1.1\%, 3.4\%]$ and $[1.3\%, 3.7\%]$, respectively. This gives a good indication of the high-quality of output from this algorithm even on seemingly unrelated measures.




**(a)** Plot of balanced score.

**(b)** Plot of sex-equal score.

**(c)** Plot of cost.

**(d)** Plot of degree.

**(e)** Plot of regret-equality score.

**(f)** Plot of regret sum.

**Figure 2** Plots of experiments to compare six different optimal stable matchings (balanced, sex-equal, egalitarian, min-regret, regret-equal, min-regret sum), and output from Algorithm REDI, over a range of measures (including balanced score, sex-equal score, cost, degree, regret-equality score, regret sum). A second order polynomial model has been assumed for all best-fit lines.

## 6 Future work

We introduced two new notions of fair stable matchings for SMI, namely, the regret-equal stable matching and the min-regret sum stable matching. We presented algorithms that are able to compute matchings of these types in polynomial time: $O(d_0nm)$ time for the regret-equal stable matching, where $d_0 = |d_U(M_0) - d_W(M_0)|$; and $O(d_sm)$ time for the min-regret sum stable matching, where $d_s = d_U(M_z) - d_U(M_0)$. It remains open as to whether these time complexities can be improved.

### References


1. D.J. Abraham, R.W. Irving, and D.F. Manlove. Two algorithms for the Student-Project allocation problem. *Journal of Discrete Algorithms*, 5(1):79–91, 2007.
2. P. Biró, J. van de Klundert, D. Manlove, W. Pettersson, T. Andersson, L. Burnapp, P. Chromy, P. Delgado, P. Dworczak, B. Haase, A. Hemke, R. Johnson, X. Klimentova, D. Kuypers, A. Nanni Costa, B. Smeulders, F. Spieksma, M.O. Valentín, and A. Viana. Modelling and optimisation in european kidney exchange programmes. *European Journal of Operational Research*, 13(4):1–10, 2019.
3. F. Cooper and D.F. Manlove. Algorithms for new types of fair stable matchings. Technical Report 2001.10875, Computing Research Repository, Cornell University Library, 2020. Available from `https://arxiv.org/abs/2001.10875`.
4. T. Feder. *Stable Networks and Product Graphs*. PhD thesis, Stanford University, 1990. Published in *Memoirs of the American Mathematical Society*, vol. 116, no. 555, 1995.
5. D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.
6. D. Gale and M. Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11:223–232, 1985.
7. S. Gupta, S. Roy, S. Saurabh, and M. Zehavi. Balanced stable marriage: How close is close enough? In *Proceedings of WADS '19: the 16th Algorithms and Data Structures Symposium*, Lecture Notes in Computer Science, pages 423–437. Springer, 2019.
8. D. Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing*, 16(1):111–128, 1987.
9. D. Gusfield and R.W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, 1989.
10. R.W. Irving and P. Leather. The complexity of counting stable marriages. *SIAM Journal on Computing*, 15(3):655–667, 1986.
11. R.W. Irving, P. Leather, and D. Gusfield. An efficient algorithm for the "optimal" stable marriage. *Journal of the ACM*, 34(3):532–543, 1987.
12. A. Kato. Complexity of the sex-equal stable marriage problem. *Japan Journal of Industrial and Applied Mathematics*, 10:1–19, 1993.
13. D.E. Knuth. *Mariages Stables*. Les Presses de L'Université de Montréal, 1976. English translation in *Stable Marriage and its Relation to Other Combinatorial Problems*, volume 10 of CRM Proceedings and Lecture Notes, American Mathematical Society, 1997.
14. D.F. Manlove. *Algorithmics of Matching Under Preferences*. World Scientific, 2013.
15. E. McDermid and R.W. Irving. Sex-equal stable matchings: Complexity and exact algorithms. *Algorithmica*, 68:545–570, 2014.
16. E. Peranson and R.R. Randlett. The NRMP matching algorithm revisited: Theory versus practice. *Academic Medicine*, 70(6):477–484, 1995.
17. O. Tange. GNU parallel - the command-line power tool. *The USENIX Magazine*, pages 42–47, 2011.